Assignment 1: Ensure the script checks if a specific file (e.g., myfile.txt) exists in the current directory. If it exists, print "File exists", otherwise print "File not found".

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ ls
  find.sh   if.sh   myfile.txt

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ cat find.sh
if [ -f "myfile.txt" ];
then
        echo "File exist"
else
        echo "$name not found"
fi

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ vim find.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ ./find.sh
File exist
```

Assignment 2: Write a script that reads numbers from the user until they enter '0'. The script should also print whether each number is odd or even.

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ ./second.sh
Enter a number:
5
./second.sh: line 13: [5: command not found
5 is odd
Enter a number:
6
./second.sh: line 13: [6: command not found
6 is even
Enter a number:
0
Exiting

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ cat second.sh

#!/bin/bash

while true; do
    echo "Enter a number:"
    read number

    if [ $number -eq 0 ]; then
        echo "Exiting"
        break
    fi

    if [ $((number % 2)) -eq 0 ]; then
        echo "$number is even"
    else
        echo "$number is odd"
    fi
Done
```

Assignment 3: Create a function that takes a filename as an argument and prints the number of lines in the file. Call this function from your script with different filenames.

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ vim third.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ ./third.sh
Number of lines in myfile.txt : 0
Number of lines in second.sh : 19
Number of lines in third.sh : 8

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ cat third.sh
countLines(){
        local filename="$1"
        local lines=$(wc -l <"$filename")
        echo "Number of lines in $filename : $lines"
}
countLines "myfile.txt"
countLines "second.sh"
countLines "third.sh"
```

Assignment 4: Write a script that creates a directory named TestDir and inside it, creates ten files named File1.txt, File2.txt, ... File10.txt. Each file should contain its filename as its content (e.g., File1.txt contains "File1.txt").

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ touch fourth.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ vim fourth.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ cat fourth.sh
mkdir TestDir
cd TestDir

for((i=1;i<=10;i++))
do
        name="File${i}"
        echo "$name">"$name"
done

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ ./fourth.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ cd TestDir

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript/TestDir
$ ls
File1  File10  File2  File3  File4  File5  File6  File7  File8  File9

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript/TestDir
$ cat file1
File1
```

Assignment 5: Modify the script to handle errors, such as the directory already existing or lacking permissions to create files.
Add a debugging mode that prints additional information when enabled.

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ touch fifth.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ vim fifth.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ ./fifth.sh
Error:Directory Already Exists

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ cat fifth.sh
if [ "$DEBUG" = "true" ]; then
        set -x
fi

errorHandler(){
        echo "Error:$1"
        exit 1
}
if [ -d "TestDir" ]; then
        errorHandler "Directory Already Exists"
fi
mkdir -p TestDir || errorHandler "Failed to create Directory"
cs TestDir || errorHandler "Failed to change Directory"
for(( i=1;i<=10;i++ )); do
        echo "File$i.txt">"File$1.txt"||errorHandler "Failed to
create 'File$i.txt'"
done
if [ "$DEBUG"="true" ]; then
        set+x
fi
```

Assignment 6: Given a sample log file, write a script using grep to extract all lines containing "ERROR". Use awk to print the date, time, and error message of each extracted line.
Data Processing with sed

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ touch sixth.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ vim sixth.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ touch Sample.log

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ vim Sample.log

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ cat Sample.log
2024-05-10 10:30:05 INFO: Application started
2024-05-10 10:30:10 ERROR: Database connection failed
2024-05-10 10:30:15 DEBUG: Processing request
2024-05-10 10:30:20 ERROR: Invalid input received
```

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ cat sixth.sh
logFile="sample.log"
grep "ERROR" "$logFile" | awk '{print $1, $2, substr($0,
index($0,$3))}' | sed 's/^[^ ] * //'


Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ ./sixth.sh
2024-05-10 10:30:10 ERROR: Database connection failed
2024-05-10 10:30:20 ERROR: Invalid input received
```

Assignment 7: Create a script that takes a text file and replaces all occurrences of "old_text" with "new_text". Use sed to perform this operation and output the result to a new file.

```
Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ touch seventh.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ vim seventh.sh

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ touch input.txt

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ vim input.txt

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ cat input.txt
this is the text in input file

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ cat seventh.sh
if [ "$#" -ne 3 ]; then
    echo "Usage: $0 <givenFile> <oldText> <newText>"
    exit 1
fi

givenFile="$1"
oldText="$2"
newText="$3"
finalFile="${givenFile%.txt}_modified.txt"
sed "s/$oldText/$newText/g" "$givenFile" > "$finalFile"

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ ./seventh.sh input.txt oldText newText

Administrator@DESKTOP-TIC5DM4 MINGW64 /shellscript
$ cat  input_modified.txt
this is the text in input file
```